An efficient method for computing steady state solutions with Gillespie's direct method

S. Mauch[a] and M. Stalzer
Center for Advanced Computing Research, California Institute of Technology, Pasadena, California 91125, USA

(Received 8 May 2010; accepted 25 August 2010; published online 12 October 2010)

Gillespie's direct method is a stochastic simulation algorithm that may be used to calculate the steady state solution of a chemically reacting system. Recently the all possible states method was introduced as a way of accelerating the convergence of the simulations. We demonstrate that while the all possible states (APS) method does reduce the number of required trajectories, it is actually much slower than the original algorithm for most problems. We introduce the elapsed time method, which reformulates the process of recording the species populations. The resulting algorithm yields the same results as the original method, but is more efficient, particularly for large models. In implementing the elapsed time method, we present robust methods for recording statistics and empirical probability distributions. We demonstrate how to use the histogram distance to estimate the error in steady state solutions. © *2010 American Institute of Physics*. [doi:10.1063/1.3489354]

I. INTRODUCTION

Consider a chemically reacting system with N species $\{S_1,\ldots,S_N\}$ and M reaction channels $\{R_1,\ldots,R_M\}$. The dynamical state of the system is denoted with the vector $X=(X_1(t),\ldots,X_N(t))$, where $X_i(t)$ is the number of molecules of S_i at time t. The reaction channel R_j is characterized by the propensity function a_j and the state change vector $v_j=(v_{1j},\ldots,v_{Nj})$. The probability that the channel fires once in an infinitesimal time $[t,t+dt)$ is $a_j dt$. The state change vector gives the change in the species populations produced by firing the reaction and is the difference between the products and reactants. Let the system satisfy the initial condition $X(t_0)=x_0$. The probability that the system will be in the state $X(t)=x$ at some later time is denoted $P(x,t|x_0,t_0)$. This joint probability distribution satisfies the chemical master equation (CME).

$$\frac{\partial P(x,t|x_0,t_0)}{\partial t}=\sum_{j=1}^{M}\left(a_j(x-v_j)P(x-v_j,t|x_0,t_0)-a_j(x)P(x,t|x_0,t_0)\right). \quad (1)$$

If the limit $P_{\mathrm{eq}}(x)=\lim_{t\to\infty}P(x,t|x_0,t_0)$ exists for all x, then $P_{\mathrm{eq}}(x)$ defines the steady state, or equilibrium, solution for the given initial condition. For systems with a finite number of states, the states may be enumerated. Then one may define a probability vector π which specifies the probability of each state occurring in the steady state solution. The finite state projection method[1] may be used to solve the CME if the number of states is not too large, or if the system may be approximated using a subset of the states that is not too large. Alternatively, one may use the optimal enumeration algorithm[2] to determine the steady state solution of the CME. The method is applicable to systems in which the copy numbers are small, and the system is either closed, or the net number of synthesis reactions is bounded. Both of these methods utilize a matrix that defines the probability of transitions between the states.

The number of possible states for a system may be very large. For the sake of simplicity, consider a system in which each of N species could have any of V possible population values. Then the number of possible states in a solution could be as large as V^N. For most systems, analytical solutions of the CME are intractable. Direct numerical solutions are possible in certain cases, but are only feasible if the number of states is small enough. Alternatively, one can numerically determine $P(x,t|x_0,t_0)$ by sampling realizations of the system. Gillespie's direct method[3,4] is a Monte Carlo algorithm for the stochastic simulation of chemical kinetics. It is used to generate trajectories, which are exact realizations of the stochastic process. At each step a reaction channel is selected using a discrete random number generator, and a time step is calculated by generating a deviate with an exponential distribution. The direct method may be used to study either transient or steady state behavior. Instead of directly recording the states, one typically collects statistics (mean, variance, and histograms) for each species. Thus the storage requirements are typically modest.

There are two methods of determining a steady state solution through Monte Carlo simulation: ensemble averaging and time averaging. For the former, one examines the limit as $t\to\infty$ of an ensemble of trajectories. The average of states across the ensemble converges to the steady state solution as the size of the ensemble increases. Of course one cannot actually carry the simulations out to infinity; one must choose a suitably large time T. Insight into the dynamics or experimentation may be needed to determine an appropriate value. With time averaging one follows a single trajectory and records each state with a weight that is equal to the time spent in that state. As $t\to\infty$, the normalized probabilities

[a]Electronic mail: sean@caltech.edu.

converge to the steady state solution. Again, one cannot follow the trajectory out to infinity, one must choose an appropriate cutoff time. In practice, one advances the simulation without recording the state to some time T_0 in order to allow the system to equilibrate. Then one records the time-weighted state up to a large fixed time T_1. Although the two methods are theoretically equivalent, time averaging is far more efficient than ensemble averaging. The reason is that in ensemble averaging an entire trajectory yields only a single data point, namely, the state at $t=T$. With time averaging, each step of the simulation yields a data point. In implementations that utilize concurrency, the two approaches are often combined. Each processor performs time averaging on an independent trajectory. At completion, the solutions are merged.

Both direct numerical methods (such as the finite state projection method and the optimal enumeration algorithm) and Monte Carlo methods (such as Gillespie's direct method) are useful tools. Of course, direct numerical solutions are preferred if the calculation is possible. The advantage of Monte Carlo methods is that they may be applied to large problems. Furthermore, the accuracy of the solution depends on the number of trajectories and perhaps the time limits. One may perform a modest calculation to obtain a rough approximation, or invest more computational effort to obtain a more accurate result.

Stochastic simulation methods are usually much more computationally expensive than deterministic methods, which typically involve numerically integrating a system of ordinary differential equations. Accurately determining the probability distribution of the species populations may require simulating many reaction events and/or generating many trajectories. Thus, much work has been done to optimize stochastic simulation algorithms. Gibson and Bruck developed the next reaction method in order to efficiently simulate systems with many reaction channels. There have been many optimizations of the direct method, most of which focus on improving the generation of discrete deviates, which are used to pick the reaction channel. The authors analyzed the performance of many formulations of stochastic simulation algorithms in Ref. 5. It has been debated which is faster, the direct method or the next reaction method. The authors found that the speed of the different formulations may differ greatly. However, in comparing the direct method and the next reaction method, one usually finds little performance difference between the best formulation of each.

In addition to optimizing the stochastic simulation algorithms themselves, there has been work in reducing the number of trajectories required to reach a certain level of accuracy. Lipshtat introduced the all possible states (APS) method[6] for accelerating the convergence when determining steady state solutions. However, we will show that for most systems the APS method has little effect on the number of required trajectories and because of the computational overhead it introduces is typically much slower than the standard method.

In Sec. II we will discuss how to record the state in order to obtain statistics and probability distributions for the species populations. In the following section we will analyze the APS method and see why it may fail to accelerate determining the steady state solution. Then in Sec. IV we will present an adaptation of the direct method that offers better performance than the standard method, especially for systems with many species.

The various algorithms presented here are implemented as part of CAIN, a stochastic simulation application with a graphical user interface. It is freely available at http://cain.sourceforge.net/. There are distributions for Mac OS X©, Microsoft WINDOWS©, and Linux/Unix operating systems.

II. RECORDING THE STATE

In quantifying the steady state solution of a system, both statistics and histograms of species populations are useful. We first consider how to accurately calculate the weighted mean and variance of a species population. Note that we are dealing with weighted statistics because each event (species population value) is weighted by the time spent in that state. Consider a sample of a population $\{x_i\}$ with associated weights $\{w_i\}$. The weighted mean μ and the unbiased estimate of the population variance s^2 are often defined in terms of the first and second moments [Eq. (2)].

$$W=\sum_{i=1}^{n} w_i, \quad \mu=\frac{1}{W}\sum_{i=1}^{n} w_i x_i,$$

$$s^2=\frac{n}{(n-1)}\left(\frac{1}{W}\sum_{i=1}^{n} w_i x_i^2-\mu^2\right). \tag{2}$$

These formulas are efficient and may be implemented with an online algorithm, that is, the formulas may be updated as new elements are added to the set. Unfortunately, the formula for the variance is numerically unstable because the variance may be much smaller than either of the two sums in the formula. In this case, the precision of the difference is much less than the precision in either of the sums.

One may obtain a numerically stable formula for the variance by first calculating the weighted mean and then defining s^2 in terms of the expectation of $x-\mu$ [Eq. (3)].

$$s^2=\frac{n}{(n-1)W}\sum_{i=1}^{n} w_i(x_i-\mu)^2. \tag{3}$$

However, this formula requires two passes through the data: one to calculate the mean and one to calculate the variance. This means that all of the elements and their weights must be stored. In accumulating statistics during simulations, this is often impractical due to the large number of events.

West's algorithm[7] is the preferred method for computing the weighted mean and variance. It is an efficient, accurate method and is also an online algorithm. As events are processed, we track four quantities: the cardinality (number of events) n, the sum of the weights W, the mean μ, and the summed second centered moment $M=\Sigma_i w_i(x_i-\mu)^2$. M is used instead of the variance because it is simpler to dynamically update. It is easy to calculate the variance from M: $s^2=nM/(n-1)W$. Each of W, μ, and M may be updated with the recurrence relations in Eq. (4).

$$W_n = W_{n-1} + w_n, \quad \mu_n = \mu_{n-1} + \frac{(x_n - \mu_{n-1})w_n}{W_n},$$

$$M_n = M_{n-1} + w_n(x_n - \mu_{n-1})^2 W_{n-1}/W_n. \tag{4}$$

The formula for the sum of the weights is obvious. The formulas for the mean and M may be verified by substituting Eq. (4) into their definitions.

Initially, each of the four quantities n, W, μ, and M are zero. Below we use West's algorithm to update these quantities for a given event x with weight w. The update uses the recurrence relations in a way that minimizes costly operations such as division. (W' and R are temporary variables.)

$$n = n + 1,$$

$$W' = W + w,$$

$$R = (x - \mu)w/W',$$

$$M = M + (x - \mu)WR,$$

$$\mu = \mu + R,$$

$$W = W'.$$

The quantities may be updated through the generation of many trajectories. If multiple trajectories are computed concurrently, one can merge the results upon their completion. The cardinality, the sum of the weights, and the summed second centered moments may simply be added. To obtain the combined mean, calculate the weighted average.

Next we consider histograms which store empirical probability distributions for the species populations. We consider uniform histograms, which have the same width for each bin. One could also use histograms with nonuniform widths. However, this would significantly increase the computational cost, as recording the state is the dominant cost in determining the steady state solution. Note that, as with the mean and variance, it is best to dynamically update the histograms. One could easily exceed the available storage (memory) by recording every reaction event, and then post-processing the data.

One can describe the structure of a histogram by specifying the bin width, the lower bound, and the number of bins. However, specifying each of these quantities at the start of a simulation is problematic. To choose an appropriate bin width one would need to know the maximum possible span of each species population, which could be ten or ten billion. An effective strategy is to specify the number of bins in each histogram and then dynamically adapt each of the bin widths and the lower bounds to span the data. A histogram gives an averaged view of an empirical probability distribution; one controls the extent of the averaging with the number of bins. One may select a large number of bins to obtain an accurate solution through generating many trajectories, or one may select a small number of bins for a less accurate solution. For example, choosing 256 bins will likely be suitable for a highly resolved solution, regardless of the mean values of the populations. Another advantage of fixing the number of bins is that one needs to allocate memory for the histograms only in the initialization phase of the simulation. Further, one can pack the histograms for all of the species into a single array to improve cache utilization.

Next we consider dynamically updating a histogram. Initially the bin width is one and the lower bound is zero. The number of bins is fixed at the user specified value. When we determine the next reaction channel to fire and the time to the next reaction, we record the current state with a weight that is the time step. Together, the bin width, lower bound, and number of bins determine the span of the histogram. If the population falls within the span of the histogram, we increment the appropriate bin value by the time step. If not, we need to adjust the bin width and lower bound. In adjusting the histogram, we first determine if we can accommodate the new event by changing only the lower bound. If so, we can rebuild the histogram by shifting the array values. If not, we double the bin width (perhaps repeatedly) until the new event is included. Then the new bin values are sums of the old bin values. One can merge histograms from concurrently generated trajectories by determining an appropriate bin width and lower bound for the combined result, and then accumulating the bin values.

Note that no information is lost when adjusting the lower bound, as long as we require that it is a multiple of the bin width. Without this restriction, we would need to perform approximations to split the content of bins. There is an inherent loss of data in increasing the bin width; a larger bin width means less precise determinations of the events. However, increasing the bin width by a factor of 2 minimizes the data loss. If the new bin width was not a multiple of the old width, then we would need to split bins. Thus, doubling is the minimum acceptable increase. Note that losing data through accumulating bins is not a bad thing. The only way to avoid losing data is to fix the bin width at one and adjust the number of bins. This would be fine for small populations, but is undesirable (and may not be computationally feasible) for large populations. How would one visualize a histogram with a million bins? By choosing the number of bins, one chooses the accuracy of the resulting histogram. Finally, note that the number of bins has no effect on the measurement of the means and variances of the species populations; that is a separate calculation.

To interpret a species population histogram as an empirical probability distribution, one simply divides the bin values by their sum. (If the bin width is not unity then technically we should add the *approximate* qualifier.) For our purposes, we consider discrete distributions defined on the natural numbers $f:\mathbb{N}\rightarrow[0\cdots1]$. One can measure the distribution distance with either the total variation metric T, which uses the one-norm, or the Kolmogorov metric K. These are defined in Eq. (5) for the distributions f and g. [We use the shorthand $f_i=f(i)$.]

$$T(f,g) = \frac{1}{2}\sum_i |f_i - g_i|, \quad K(f,g) = \max_j \left| \sum_{i=0}^{j} (f_i - g_i) \right|. \tag{5}$$

Reference 8 analyzes these in the context of stochastic simulations, considering both the number of bins and the number

of recorded events. (They consider transient behavior, i.e., species populations at a specified point in time, but the results are also applicable to steady state behavior.) If one knows the exact solution, then computing the distance (either total variation or Kolmogorov) between the empirical solution and the exact solution is a measurement of the error. (Note that Cao and Petzold use the term *histogram distance* to denote twice the total variation distance, whereas we use the term to denote either the total variation distance or the Kolmogorov distance, both of which have values in the range $[0\cdots1]$.) One may use this technique to determine the accuracy of approximate stochastic simulation methods like τ-leaping[9] by comparing the empirical solution with either an analytical solution or a converged solution determined with an exact method. One may also measure the effect of changing model parameters by computing the distance between the resulting solutions.

We will consider the total variation metric for calculating the distance between histograms. The most common use of the histogram distance is in determining if one has generated enough trajectories. Since most models are analytically intractable, one must work only with empirical solutions in determining the error. Consider an empirical solution with B bins and cardinality N. Although one cannot measure the distance between the empirical solution and the exact solution, one can generate other independent empirical solutions and measure the distance between them. When one uses an exact method, such as Gillespie's direct method, the distance between two empirical solutions of the chemical master equation is a random variable called the *self distance*. This quantity may be used in place of a direct measurement to an unknown exact solution. In Ref. 8 it is shown that the mean of the self distance is bounded by $\sqrt{B/\pi N}$. That the self distance is inversely proportional to the square root of the cardinality indicates that reducing the error by a factor of r requires increasing the cardinality (number of samples) by a factor r^2. We also see that the sensitivity of the self distance is directly proportional to the square root of the number of bins. (In the limiting case of a single bin the self distance is identically zero.) Thus one might choose a small number of bins when generating a rough solution (a small number of samples) and a larger number of bins for a more detailed solution. This brings us back to the question: have we generated enough trajectories? One could simply look at the histograms. If they are smooth enough for one's taste, then the solution is sufficient. For a less *ad hoc* approach, one may sample the self distance by generating an independent solution using the same number of trajectories (and hence a similar number of samples.)

Of course, it is better still to use statistical methods to estimate the error. We consider the error in an empirical probability distribution that is stored in a set of m histograms. First we generate a number of independent trajectories that result in the histograms $h_1,\ldots,h_m$. We will denote the mean distribution (the combined result of all of the histograms) with $\bar{h}$. The combined histogram $\bar{h}$ is more accurate than any one of its components h_i. The biased estimate of the error in the ith histogram is the distance from the mean distribution $T(h_i,\bar{h})$. (Here we are using the total variation distance.) We combine these errors to obtain the average unbiased estimate of the error in each of the histograms $(1/(m-1))\Sigma_{i=1}^{m}T(h_i,\bar{h})$. Recall that the convergence rate of the metric is $1/\sqrt{N}$ where N is the cardinality. With the assumption that each of the histograms has approximately the same cardinality, the error in the mean distribution is a factor of $\sqrt{m}$ less than the average histogram error. Thus, the estimate of the error in the mean distribution $\bar{h}$ is $e=(1/(m-1)\sqrt{m})\Sigma_{i=1}^{m}T(h_i,\bar{h})$. This formula is analogous to the standard error in the mean for a random variable.

The process of generating additional solutions in order to check the error in one that has already been generated is cumbersome for a software user. Thus, it is best to automatically store the solution in a small number of independent groups of histograms. An equal number of trajectories are used to generate each group. The number of groups is the *histogram multiplicity*. Note that the results are merged to obtain the mean histogram for plotting or other analysis. While using multiple solutions increases the storage requirements, it has a negligible effect on the execution time, as each whole trajectory contributes to just one of the sets of histograms. (Also, the storage required for the histograms is not a limiting factor for most problems.) This procedure has been implemented in CAIN. The user selects the histogram multiplicity when launching a suite of simulations. The default value is four. One may choose a larger multiplicity to obtain more accurate estimates of the histogram errors, or a smaller value if one is not interested in analyzing these errors.

III. THE ALL POSSIBLE STEPS ALGORITHM

Lipshtat[6] introduced the all possible steps method as an alternative to the standard method of determining the steady state of a system with Gillespie's direct method. In the standard method, one records the state with the time step as the weight. In the APS method, one considers all possible reactions that may occur. The probability of the mth reaction firing is $a_m(\mathbf{x})/\alpha(\mathbf{x})$, where $a_m(\mathbf{x})$ is the mth reaction propensity and $\alpha(\mathbf{x})$ is the sum of the propensities. Let $\mathbf{v}_m$ be the state change vector for the mth reaction. If that reaction fires, the system will remain in the state $\mathbf{x}+\mathbf{v}_m$ for a time τ_m that is an exponential deviate with mean $1/\alpha(\mathbf{x}+\mathbf{v}_m)$. In each time step of the APS method, each reaction is fired virtually. For the mth reaction, the state $\mathbf{x}+\mathbf{v}_m$ is recorded with a weight of $a_m(\mathbf{x})\tau_m/\alpha(\mathbf{x})$, where τ_m is calculated using the same exponential deviate that was used to determine the time step. Thus, the APS method takes more samples of the state without the need for additional random deviates.

Reference 6 uses a protein dimerization model as a test problem. There are two species, S_1 and S_2, and three reactions, production $\varnothing\xrightarrow{k_1}S_1$, degradation $S_1\xrightarrow{k_2}\varnothing$, and dimerization $2S_1\xrightarrow{k_3}S_2$. The stochastic rate constants are $k_1=5$, $k_2=2$, and $k_3=4$, respectively. [Reference 6 uses the convention for deterministic rate constants, for which $k_3=2$. For this second order reaction, the deterministic rate is $k_3[X]^2$, while the stochastic propensity[10] is $k_3[X]([X]-1)/2$.] S_1 is the only reac-

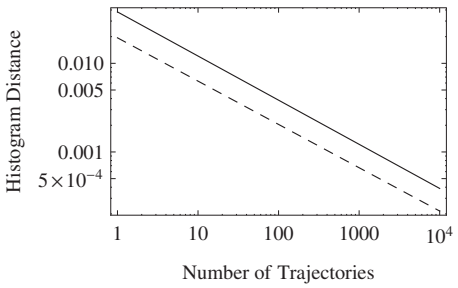

FIG. 1. The error, measured as histogram distance, as a function of the number of trajectories. The least-squares fits to the data are shown. The performance for the standard method and the APS method are plotted with a solid line and a dashed line, respectively. The stochastic rate constants are $k_1=5$, $k_2=2$, and $k_3=4$. The simulation was advanced for ten time units to allow the system to achieve steady state, then the state was recorded for 150 time units. Each test was performed 100 times. The APS method requires about 0.27 times as many trajectories as the standard method to reach a given level of accuracy.

tant species. For a steady state analysis we may ignore S_2, which does not have a steady state solution. This model has an analytical solution for S_1. Thus we use the exact steady state solution when analyzing the errors in the simulation methods.

Using the APS method (implemented as part of CAIN) allows one to converge to a given accuracy level with fewer trajectories. If one measures the relative error in the mean of S_1, then the number of required trajectories is reduced by a factor of 19. Next we use the total variation metric to measure the distance between the empirical probability distributions for S_1 and the exact probability distributions. This is a better measure of the error in the solution than the error in the mean. Figure 1 shows this error as a function of the number of trajectories. We see that the APS method reduces the number of required trajectories by a factor of about 3.8. When using the histogram distance measure of error, the reduction factor is not as large as that presented in Ref. 6, but the findings are consistent.

For the protein degradation problem, the mean value of the steady state population of S_1 is approximately 0.98. Now we reduce the stochastic rate constants for the degradation and dimerization reactions to $k_2=0.025$ and $k_3=0.05$, respectively. This increases the mean value of S_1 to about 9.9. The error, as measured with the total variation metric, for this model is shown in Fig. 2. The APS method still requires

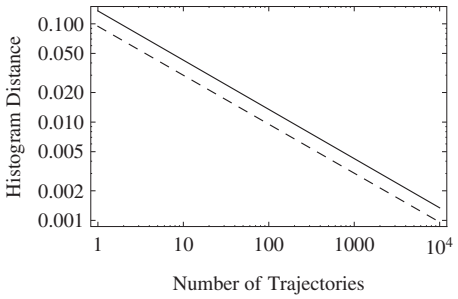

FIG. 2. The error, measured as histogram distance, as a function of the number of trajectories. The stochastic rate constants are $k_1=5$, $k_2=0.025$, and $k_3=0.05$. It takes about 0.51 times as many trajectories to achieve an error of 0.01 with the APS method than with the standard method.

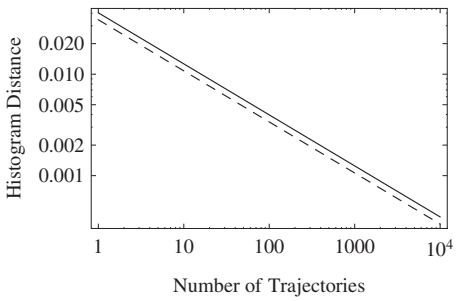

FIG. 3. The error, measured as histogram distance, as a function of the number of trajectories. The rate constants have the original values, but the system is duplicated by a factor of 10. It takes about 0.69 times as many trajectories to achieve an error of 0.01 with the APS method than with the standard method.

fewer trajectories to converge to a given accuracy, however, the fraction has risen from 0.27 to 0.51. If the stochastic rate constants are further decreased so that the mean population is approximately 100, then the fraction of required trajectories increases to 0.76. We see that performing extra sampling can significantly reduce the required number of trajectories only if the samples have significantly different values. Thus, as the average population of the species being recorded increases, there is decreased benefit in the extra sampling done in the APS method.

Of course, if one is using stochastic methods to study a model, then one would expect that at least some of the species populations are small. Otherwise the stochastic effects would not be important. However, for the APS method to significantly reduce the number of required trajectories, the populations must be very low.

We have considered the effect of the mean population. Now we consider how the number of species affects the number of required trajectories. We take the original protein dimerization model and duplicate it by a factor of 10. Thus there are 10 species and 30 reactions. We record each of the species populations in calculating the steady state solutions. Since the species are not coupled, each one has the same steady state solution as the original model. In Fig. 3 we show the error in the empirical probability distribution for the first species as a function of the number of trajectories. The fraction of required trajectories for the APS method is about 0.69, which is much higher than the fraction of 0.27 in the original model. By contrast, increasing the number of species does not affect the convergence rate for the standard method. The species are not coupled, so the behavior of each one is the same as that in the original model. The reason for the increase in required trajectories for the APS method is that for any one of the ten species 90% of the reactions have no effect on its population. This is not a peculiarity of the duplicated model. Most models with more than a few species have the property that for any particular species most reactions do not directly change that species population. This follows from the property that each reaction has a small number of reactants and products. In fact, for mass action kinetics only up to second order reactions (with two reactants or a single reactant of order two) can be considered to be realistic.[10] For our duplicated model, at each time step the APS method virtually fires each of the 30 reactions and

records the state with appropriate weights. Yet for any given species, only three of those reactions change its population. Hence most of the samples record the same value (with different weights). Thus we see that the fraction of required trajectories with the APS method increases both with the mean species populations and the number of species.

The APS method does reduce the number of trajectories that are required to reach a specified accuracy. Unless the model has very few species and very low mean populations, the effect is modest, but it is still a reduction. However, the number of required trajectories is not a practical performance measure. The *important* measure is the error as a function of the time it takes to compute the solution. For the duplicated model, the APS method requires about 0.69 times as many trajectories as the standard method. For this model, generating a trajectory with the APS method takes approximately 20 times as long as with the standard method. This indicates that reaching a specified level of accuracy with the APS method takes 14 times as long. The reason for the poor performance of the APS method is that, except for models with only a couple of species and reactions, recording the state is the dominant computational cost. This contradicts the claim in Ref. 6 that "Updating several probabilities at any step requires of course more computations. However, since these are only standard arithmetic operations, there is no significant overhead in terms of running time." For the duplicated model, an empirical histogram and population statistics are updated for each of the 30 reactions and each of the ten species. Thus at each time step, there are 300 updates. By comparison, the standard method does only ten.

Now we will analyze the asymptotic computational complexity of the component algorithms in Gillespie's direct method to see why the extra sampling in the APS method typically dominates the computational cost of the simulation. Consider determining the steady state solution for a model with N species and M reactions. In taking a time step, one must generate a discrete deviate to pick the reaction channel, generate an exponential deviate to calculate the time step, and change the populations by firing a reaction. There are efficient methods of performing each of these steps.[5] There are various algorithms for generating a discrete deviate which range in asymptotic computational complexity from $\mathcal{O}(M)$ down to $\mathcal{O}(1)$. Exponential deviates may be efficiently generated with either the ziggurat method[11] or the acceptance complement method.[12] One can efficiently update the state by using sparse arrays for the state change vectors.[13] Now consider the cost of sampling the species populations. The standard method samples each of the species, so the computational complexity of sampling is $\mathcal{O}(N)$. The APS method samples the species populations upon virtually firing each reaction channel. Because N species are recorded for each of the M reaction channels the complexity of recording the state is $\mathcal{O}(NM)$. It is clear that for large models, the cost of sampling will dominate when using the APS method. Our numerical tests have shown that for an optimized solver the cost of sampling dominates, even for the model with ten species.

Lipshtat demonstrates that one of the advantages of using the APS method in conjunction with the standard method is that the two methods yield different solutions. Measuring the difference between these two solutions gives an indication of the error in the average solution. However, using multiple histograms to record the state, as detailed in Sec. II, yields a more meaningful estimate of the error.

IV. USING THE ELAPSED TIME

The APS method is usually not efficient because it does so much sampling, but even for the standard method sampling the populations becomes the dominant cost as the number of species increases. Thus, we seek a way to reduce this cost. Again consider an abstract model that has N species and M reactions. Assume that we are determining the steady state solution for some subset of the species. Because each reaction channel has a small number of reactants and products, each affects only a few species. In models with more than a few species, for any given reaction, most species populations are unaffected. Suppose that after a reaction channel which affects a particular species X fires, there are $n-1$ steps in which X is not modified, until the nth step in which it is. Let $\{\tau_i\}$ be the time increments for these steps. X is sampled n times with different weights, but the same population value. We could instead sample X a single time with the sum of these weights. The effect in terms of the empirical probability distribution, or in terms of the statistics, is the same.

To reduce the number of times we sample each species, for each reaction we store the species that it modifies. Conceptually we have a list of lists; for each reaction there is a list of integers representing species indices. However, for efficiency this is implemented with a single packed array. We also need an array to store the time at which each species was last modified. Now we can reduce the sampling. Consider a time step. Instead of looping over the recorded species, we loop over the species that will be modified when the determined reaction channel fires at the end of the time step. For each of these species that will be affected, we sample the population with a weight that is the difference between the time at the end of the step and the time at which the species was last modified. Below we present pseudocode for a step with the elapsed time method. After determining the reaction channel and time step, the time is incremented. Then the state is recorded for the modified species. The array lm stores the times at which each species was last modified.

$\mu =$ the reaction to fire next,

$\tau =$ the time to the next reaction,

$t = t + \tau$,

for each species index n modified by reaction μ

record species n with weight $t - \mathrm{lm}[n]$,

$\mathrm{lm}[n] = t$.

To test the performance of using the elapsed time, we again consider the duplicated protein dimerization model with ten species. Previously in Fig. 3, we showed the error as a function of the number of trajectories. In Fig. 4 we consider three methods: standard, APS, and elapsed time, but

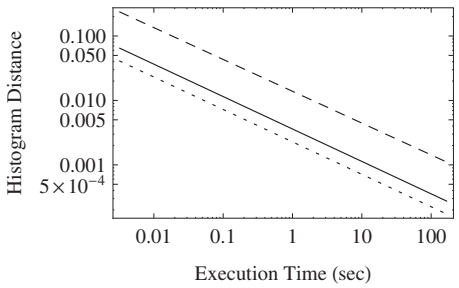

FIG. 4. The error, measured as histogram distance, as a function of the execution time in seconds. The rate constants have the original values, but the system is duplicated by a factor of 10. The performance for the standard method, the APS method, and the elapsed time method are plotted with a solid line, a dashed line, and a dotted line, respectively.

plot the error as a function of execution time. From the least-squares fits, we calculate that the APS method takes approximately 15 times as long to converge to a specified accuracy as the standard method. The elapsed time method takes about 0.4 times as long as the standard method.

Now we add reactions to the duplicated protein dimerization model to obtain a system with coupling between the different species. For each species S_i, we add the reactions $S_i \xrightarrow{k_4} S_{(i-1) \bmod N}$ and $S_i \xrightarrow{k_5} S_{(i+1) \bmod N}$, each with a stochastic rate constant of 2. For a model with N species, there are $5N$ reactions. We determine the equilibrium solutions as before. For a multiplicity of 10, the elapsed time method takes 0.65 times as long to converge to a specified accuracy as the standard method. The APS method takes 22 times as long as the standard method. For a multiplicity of 100, the performance differences widen. The elapsed time method and the APS method take 0.16 and 270 times as long as the standard method, respectively.

V. CONCLUSIONS

When calculating the mean and variance of species populations, it is best to use robust methods such as West's algorithm.[7] With this approach, one may efficiently maintain these statistics in serial or concurrent simulations. In recording empirical probability distributions for species populations, it is convenient to fix the number of bins in the histogram, and dynamically change the bin width and lower bound to capture the recorded data. If one records the state for a given species in multiple histograms, then one may estimate the error in the resulting probability distribution by using the histogram distance from the combined distribution.

The all possible states method reduces the number of trajectories required to reach a specified error tolerance in species probability distributions. The reduction factor decreases both with increasing mean population and increasing number of species. Thus, the factor is only significant when there are few species, each with low populations. The computational complexity of a time step with the APS method is the product of the number of species and number of reactions. Thus, generating a trajectory with the APS method is more expensive than doing so with the standard method. The relative expense increases with the number of species and reactions. As a result, the APS method is typically much slower than the standard method.

The elapsed time method reformulates the process of recording the species populations. It yields the same results as the standard method, but reduces the costs of updating the statistics and histograms. For small problems, the elapsed time method typically gives a modest reduction in execution time. For models with many species, the cost of recording the state dominates, and the elapsed time method is much more efficient than the standard method.

ACKNOWLEDGMENTS

This project was supported by Grant No. R01EB007511 from the National Institute of Biomedical Imaging and Bioengineering. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of Biomedical Imaging and Bioengineering or the National Institutes of Health. The authors gratefully acknowledge Dan Gillespie, Linda Petzold and her research group at UCSB, Michael Hucka, and John McCorquodale for many useful conversations and comments.

[1] B. Munsky and M. Khammash, J. Chem. Phys. **124**, 044104 (2006).
[2] Y. Cao and J. Liang, BMC Syst. Biol. **2**, 30 (2008).
[3] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[4] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[5] S. Mauch and M. Stalzer, "Efficient formulations for exact stochastic simulation of chemical systems," IEEE/ACM Trans. Comput. Biol. Bioinf. (in press).
[6] A. Lipshtat, J. Chem. Phys. **126**, 184103 (2007).
[7] D. H. D. West, Commun. ACM **22**, 532 (1979).
[8] Y. Cao and L. Petzold, J. Comput. Phys. **212**, 6 (2006).
[9] D. T. Gillespie, Annu. Rev. Phys. Chem. **58**, 35 (2007).
[10] D. J. Wilkinson, *Stochastic Modelling for Systems Biology* (CRC, Boca Raton, FL, 2006).
[11] G. Marsaglia and W. W. Tsang, J. Stat. Software **5**, 1 (2000).
[12] H. Rubin and B. Johnson, J. Stat. Comput. Simul. **76**, 509 (2006).
[13] M. A. Gibson and J. Bruck, J. Phys. Chem. A **104**, 1876 (2000).